Visual servoing and pose estimation with cameras obeying the unified model

Omar Tahri, Youcef Mezouar, François Chaumette, Helder Araujo

▶ **To cite this version:**

Omar Tahri, Youcef Mezouar, François Chaumette, Helder Araujo. Visual servoing and pose estimation with cameras obeying the unified model. Chesi, G. and Hashimoto, K. Visual Servoing via Advanced Numerical Methods, LNCIS 401, Springer-Verlag, pp.231–252, 2010. inria-00548936

HAL Id: inria-00548936
https://inria.hal.science/inria-00548936

Submitted on 20 Dec 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Chapter 13
Visual Servoing and Pose Estimation with Cameras Obeying the Unified Model

Omar Tahri, Youcef Mezouar, François Chaumette, and Helder Araujo

Abstract In this chapter, both visual servoing and pose estimation from a set of points are dealt with. More precisely, a unique scheme based on the projection onto the unit sphere for cameras obeying the unified model is proposed. From the projection onto the surface of the unit sphere, new visual features based on invariants to rotations are proposed. It is shown that satisfactory results can be obtained using these features for visual servoing and pose estimation as well.

13.1 Introduction

Visual servoing aims at controlling robotic systems by the information provided by one or more cameras. According to the space where the visual features are defined, several kinds of visual servoing can be distinguished. In position-based visual servoing (PBVS) [33, 21], the features are defined in the 3D space. An adequate 3D trajectory is usually obtained using PBVS, such as a geodesic for the orientation and a straight line for the translation. However, position-based visual servoing may suffer from potential instabilities due to image noise [5]. Furthermore, the knowledge of an exact object 3D model is required. On the contrary, in image-based visual servo (IBVS) [9], the robot motions are controlled by canceling errors on visual features defined in the image. This kind of visual-servo is more robust to image noise and calibration errors than PBVS, in general. However, as soon as the initial

Omar Tahri and Helder Araujo
Institute for Systems and Robotics, Polo II 3030-290 Coimbra, Portugal, e-mail: {omartahri, helder}@isr.uc.pt

Youcef Mezouar
LASMEA, University Blaise Pascal, Campus des Cezeaux, 63177 Aubiere, France, e-mail: mezouar@univ-bpclermont.fr

François Chaumette
INRIA, Campus de Beaulieu, 35042 Rennes, France, e-mail: francois.chaumette@irisa.fr

error between the initial and the desired poses is large, the 3D behavior becomes unpredictable when the visual features are not adequately chosen. Furthermore, other problems may appear such as reaching local minimum or a task singularity [5]. A compromise and an hybrid visual servoing [19] can be obtained by combining features in image and partial 3D data.

In this chapter, we are concerned with IBVS. In fact, the main cause of trouble for IBVS is the strong nonlinearities in the relation from the image space to the workspace that are generally observed in the interaction matrix. In principle, an exponential decoupled decrease will be obtained simultaneously on the visual features and on the camera velocity, which would provide a perfect behavior, if the interaction matrix is constant. Unfortunately, that is not usually the case. To overcome the problem of nonlinearities observed in the interaction matrix, an approach consists in using the measures to build particular visual features that will ensure expected properties of the control scheme. In fact, the way to design adequate visual features is directly linked to the modeling of their interaction with the robot motion, from which all control properties can be analyzed theoretically. If the interaction is too complex (i.e. highly nonlinear and coupled), the analysis becomes impossible. Several works have been realized in IBVS following this general objective. In [24], a vanishing point and the horizon line have been selected. This choice ensures a good decoupling between translational and rotational degrees of freedom (DOF). In [15], vanishing points have also been used for a dedicated object (a 3D rectangle), once again for decoupling properties. For the same object, six visual features have been designed in [6] to control the 6 DOF of a robot arm, following a partitioned approach. In [14], the coordinates of points are expressed in a cylindrical coordinate system instead of the classical Cartesian one, so as to improve the robot trajectory. In [13], the three coordinates of the centroid of an object in a virtual image obtained through a spherical projection have been selected to control 3 DOF of an underactuated system. Recently, [11] proposed a decoupled visual servoing from spheres using a spherical projection model. Despite of the large quantity of results obtained in the last few years, the choice of the set of visual features to be used in the control scheme is still an open question in terms of stability analysis and validity for different kinds of sensor and environment.

In this chapter, invariants computed from the projection onto the surface of the unit sphere will be used to improve the IBVS behavior in terms of convergence domain and 3D behavior. In previous works, the invariance property of some combinations of image moments computed from image regions or a set of points have been used to decouple the DOF from each-other. For instance, in [28, 30], moments allow using of intuitive geometrical features, such as the center of gravity or the orientation of an object. However, these works only concerned planar objects and conventional perspective cameras. More recently, a new decoupled IBVS from the projection onto the unit sphere has been proposed in [31]. The proposed method is based on polynomials invariant to rotational motion computed from a set of image points. This current work improves the proposed features given in [31]. More precisely, the new features allow obtaining interaction matrices almost constant with

respect to the depth distributions. This decreases the system nonlinearity and improves the convergence speed and rate.

The second part of this chapter deals with the pose estimation problem. There are many applications of pose estimation, where the 6 parameters of the camera pose have to be calculated from known correspondences with known scene structure: robot localization using a vision sensor, or PBVS [33]. The pose estimation is one of most classical problem in vision [7, 16]. This problem is more than 150 years old and there is recent renewed interest because of automated navigation and model-based vision systems. Numerous methods have been proposed in the literature and giving an exhaustive list of them is certainly impossible. The pose estimation methods can be divided into several categories according to the used features, direct methods or iterative methods. The geometric features considered for the estimation of the pose are often points [7], segments [8], contours, conics [25] or image moments [29]. Another important issue is the registration problem. Purely geometric [8], or numerical and iterative [7] approaches may be considered. Linear approaches are suitable for real-time applications and give closed-form solutions free of initialization [10, 1]. Full-scale nonlinear optimization techniques [17] consist of minimizing the error between the observation and the projection of the model. The main advantage of these approaches is their accuracy. The main drawback is that they may be subject to local minima and, worse, divergence.

The method we propose in this chapter is based on virtual visual servoing (VVS) using moment invariants as features. In other words, we consider the problem of the pose computation as similar to the positioning of a virtual camera using features in the image [26, 20]. This method is equivalent to nonlinear methods that consist in minimizing a cost function using iterative algorithms. The main idea behind the method we propose is based on the following fact: the features that can be used for visual servoing to ensure a large convergence domain and adequate 3D behavior can be used to obtain a large convergence domain and high convergence speed for pose estimation using VVS.

As mentioned above, the features we propose are computed from the projection onto the unit sphere. This means that the proposed method can be applied not only to conventional cameras but also to all omnidirectional cameras obeying the unified model [12, 4]. Omnidirectional cameras are usually intended as a vision system providing a 360^o panoramic view of the scene. Such an enhanced field of view can be achieved by either using catadioptric systems, obtained by simply combining mirrors and conventional cameras, or employing purely dioptric fisheye lenses [3]. In practice, it is highly desirable that such imaging systems have a single viewpoint [3, 27]. That is, there exists a single center of projection, so that, every pixel in the sensed images measures the irradiance of the light passing through the same viewpoint in one particular direction. The reason why a single viewpoint is so desirable is that it permits the extension of several results obtained for conventional cameras. The pose estimation method we propose is thus valid for catadioptric, conventional and some fisheye cameras.

The following of this chapter is organized as follow:

- in the next section, the unified camera model is recalled;

- in Section 13.3, the theoretical background of this work is detailed;
- in Section 13.4, the feature choice to control the 6 DOF of the camera or to estimate its pose is explained;
- in Section 13.5, validations results for visual servoing and pose estimation are presented. In this way, the pose estimation method using VVS is compared to linear pose estimation method [1] and an iterative method [2].

13.2 Camera Model

Central imaging systems can be modeled using two consecutive projections: spherical then perspective. This geometric formulation called the *unified model* was proposed by Geyer and Daniilidis in [12]. Consider a virtual unitary sphere centered on C_m and the perspective camera centered on C_p (refer to Fig. 13.1). The frames attached to the sphere and the perspective camera are related by a simple translation of $-\xi$ along the Z-axis. Let $\mathcal{X}$ be a 3D point with coordinates $\mathcal{X} = (X, Y, Z)^\top$ in $\mathcal{F}_m$. The world point $\mathcal{X}$ is projected onto the image plane at a point with homogeneous coordinates $\mathbf{p} = \mathbf{Km}$, where $\mathbf{K}$ is a 3×3 upper triangular matrix containing the conventional camera intrinsic parameters coupled with mirror intrinsic parameters and

$$\mathbf{m} = \Big(x,\, y,\, 1\Big)^\top = \Big(\tfrac{X}{Z+\xi\|\mathcal{X}\|},\, \tfrac{Y}{Z+\xi\|\mathcal{X}\|},\, 1\Big)^\top. \tag{13.1}$$

The matrix $\mathbf{K}$ and the parameter ξ can be obtained after calibration using, for example, the methods proposed in [22]. In the sequel, the imaging system is assumed to be calibrated. In this case, the inverse projection onto the unit sphere can be obtained by:

$$\mathcal{X}_\mathbf{s} = \lambda\Big(x,\, y,\, 1 - \tfrac{\xi}{\lambda}\Big)^\top, \tag{13.2}$$

where $\lambda = \frac{\xi + \sqrt{1+(1-\xi^2)(x^2+y^2)}}{1+x^2+y^2}$

Note that the conventional perspective camera is nothing but a particular case of this model (when $\xi = 0$). The projection onto the unit sphere from the image plane is possible for all sensors obeying the unified model.

13.3 Mathematical Background

13.3.1 Visual Servoing and Pose Estimation

In few words, we recall that the time variation $\dot{\mathbf{s}}$ of the visual features $\mathbf{s}$ can be expressed linearly with respect to the relative camera-object kinematics screw $\mathbf{V} = (\mathbf{v}, \boldsymbol{\omega})$:

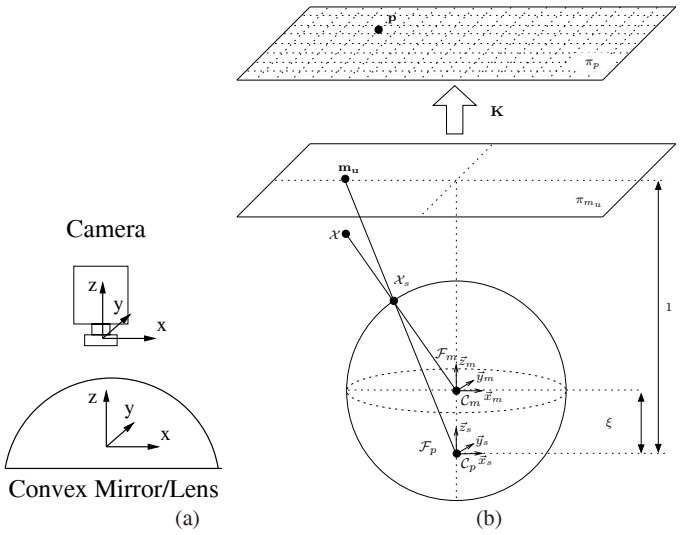

Fig. 13.1 (a) axis convention; and (b) unified image formation.

$$\dot{\mathbf{s}} = \mathbf{L_s}\mathbf{V}, \tag{13.3}$$

where $\mathbf{L_s}$ is the interaction matrix related to $\mathbf{s}$. The control scheme is usually designed to reach an exponential decoupled decrease of the visual features to their desired value $\mathbf{s}^*$. If we consider an eye-in-hand system observing a static object, the control law is defined as follow:

$$\mathbf{V_c} = -\lambda \widehat{\mathbf{L_s}}^{+}(\mathbf{s} - \mathbf{s}^*), \tag{13.4}$$

where $\widehat{\mathbf{L_s}}$ is a model or an approximation of $\mathbf{L_s}$, $\widehat{\mathbf{L_s}}^{+}$ the pseudo-inverse of $\widehat{\mathbf{L_s}}$, λ a positive gain tuning the time to convergence, and $\mathbf{V_c}$ the camera velocity sent to the low-level robot controller. The nonlinearities in system (13.4) explain the difference of behaviors in image space and in 3D space, and the inadequate robot trajectory that occurs sometimes when the displacement to realize is large (of course, for small displacements such that the variations of $\widehat{\mathbf{L_s}}$ are negligible, a correct behavior is obtained). An important issue is thus to determine visual features allowing to reduce the nonlinearities in (13.4). Furthermore, using (13.4) local minima can be reached when the number of features is not minimal. Therefore, one would like to chose a minimal representation (the number of features is equal to the number of DOF), but without singularities and robust with respect to image noise.

The problem of pose estimation consists in determining the rigid transformation ${}^c\mathbf{M}_o$ between the object frame $\mathcal{F}_o$ and the camera frame $\mathcal{F}_c$ in unknown position using the corresponding object image (see Fig. 13.2). It is well known that the relation between an object point with coordinates $\mathcal{X}_c = (X_c, Y_c, Z_c, 1)$ in F_c and $\mathcal{X}_o = (X_o, Y_o, Z_o, 1)$ in F_o can be written:

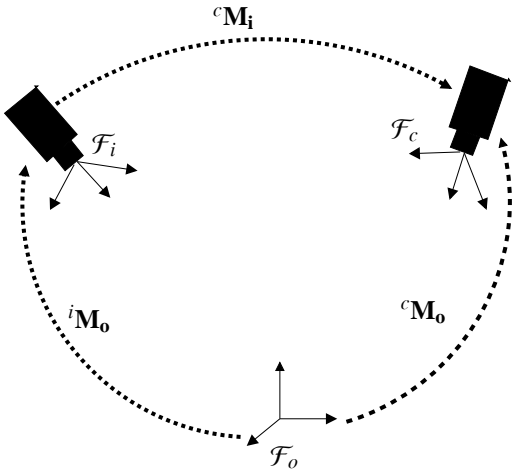

Fig. 13.2 Pose estimation using VVS.

$$\mathcal{X}_c = {}^c\mathbf{M}_o \mathcal{X}_o = \begin{bmatrix} {}^c\mathbf{R}_o & {}^c\mathbf{t}_o \\ \mathbf{0_{31}} & 1 \end{bmatrix} \mathcal{X}_o. \tag{13.5}$$

The matrix ${}^c\mathbf{M}_o$ can be estimated by minimizing the error module in image:

$$e = \| \mathbf{s}({}^c\mathbf{M}_o) - \mathbf{s}^* \|, \tag{13.6}$$

where $\mathbf{s}^*$ is the value of a set of visual features computed in the image acquired in the camera unknown position and $\mathbf{s}(^\mathbf{c}\mathbf{M_o})$ is the value of the same set of features computed from the object model, the transformation ${}^c\mathbf{M}_o$, and the camera model. VVS consists in moving a virtual camera from a known initial pose ${}^i\mathbf{M}_o$ (referenced by the frame $\mathcal{F}_i$ on Fig. 13.2) to the final unknown pose (referenced by the frame $\mathcal{F}_c$ on Fig. 13.2) where e is minimized. In fact, the main difference between VS and VVS is that the visual features at each iteration are computed in VVS, while they are extracted from acquired images in VS. However, the displacement of the camera (real or virtual) is computed using the same control law (13.4).

13.3.2 Moments from the Projection of Points onto the Surface of Unit Sphere

13.3.2.1 Definitions

The 3D moment of order $i + j + k$ computed from a discrete set of N points are defined by the following classical equation:

$$m_{i,j,k} = \sum_{h=1}^{N} x_{s_h}^i \, y_{s_h}^j \, z_{s_h}^k, \tag{13.7}$$

where $(x_{s_h}, y_{s_h}, z_{s_h})$ are the coordinates of a 3D point. In our application, these coordinates are nothing but the coordinates of a point projected onto the unit sphere. They can be computed from the projection of a point onto the image plane and the inverse transform (13.2).

13.3.2.2 Interaction Matrix

In the case of moments computed from a discrete set of points, the derivative of (13.7) with respect to time is given by:

$$\dot{m}_{i,j,k} = \sum_{h=0}^{N} (i \, x_{s_h}^{i-1} \, y_{s_h}^j \, z_{s_h}^k \dot{x}_{s_h} + j \, x_{s_h}^i \, y_{s_h}^{j-1} \, z_{s_h}^k \dot{y}_{s_h} + k \, x_{s_h}^i \, y_{s_h}^j \, z_{s_h}^{k-1} \dot{z}_{s_h}). \tag{13.8}$$

For any set of points (coplanar or noncoplanar), the interaction matrix related to $\mathbf{L}_{\mathbf{m_{i,j,k}}}$ can thus be obtained by combining (13.8) with the well known interaction matrix $\mathbf{L}_{\mathcal{X}_\mathbf{s}}$ of a point $\mathcal{X}_s$ on the unit sphere (defined such that $\dot{\mathcal{X}}_\mathbf{s} = \mathbf{L}_{\mathcal{X}_\mathbf{s}}\mathbf{V}$) [13, 28, 11]:

$$\mathbf{L}_{\mathcal{X}_\mathbf{s}} = \left[-\tfrac{1}{r}\mathbf{I_3} + \tfrac{1}{r}\mathcal{X}_s\mathcal{X}_s^\top \quad [\mathcal{X}_s]_\times \right], \tag{13.9}$$

where r is the distance of the 3D point to the sphere center. In the particular case of a coplanar set of points, the interaction matrix related to $m_{i,j,k}$ can be determined [28]:

$$\mathbf{L}_{m_{i,j,k}} = \left[m_{vx} \; m_{vy} \; m_{vz} \; m_{wx} \; m_{wy} \; m_{wz} \right], \tag{13.10}$$

where:

$$\begin{cases} m_{vx} = A(\beta_d m_{i+2,j,k} - i m_{i,j,k}) \\ \qquad + B(\beta_d m_{i+1,j+1,k} - i m_{i-1,j+1,k}) \\ \qquad + C(\beta_d m_{i+1,j,k+1} - i m_{i-1,j,k+1}), \\ m_{vy} = A(\beta_d m_{i+1,j+1,k} - j m_{i+1,j-1,k}) \\ \qquad + B(\beta_d m_{i,j+2,k} - j m_{i,j,k}) \\ \qquad + C(\beta_d m_{i,j+1,k+1} - j m_{i,j-1,k+1}) \\ m_{vz} = A(\beta_d m_{i+1,j,k+1} - k m_{i+1,j,k-1}) \\ \qquad + B(\beta_d m_{i,j+1,k+1} - k m_{i,j+1,k-1}) \\ \qquad + C(\beta_d m_{i,j,k+2} - k m_{i,j,k}), \\ m_{wx} = j m_{i,j-1,k+1} - k m_{i,j+1,k-1} \; , \\ m_{wy} = k m_{i+1,j,k-1} - i m_{i-1,j,k+1} \; , \\ m_{wz} = i m_{i-1,j+1,k} - j m_{i+1,j-1,k} \; , \end{cases}$$

with $\beta_d = i + j + k$ and (A, B, C) are the parameters defining the object plane in the camera frame:

$$\frac{1}{r} = \boldsymbol{\alpha}^\top \mathcal{X}_s = Ax_s + By_s + Cz_s. \tag{13.11}$$

13.4 Features Choice

In this section, the features choice is detailed. We will first explain how to obtain features to control the translational DOF with interaction matrices almost constant with respect to depth distributions. Then, a vector of features to control the whole 6 DOF will be proposed.

13.4.1 Invariants to Rotational Motion

The shape of an object does not change under rotational motions. After a rotational motion of the camera frame, it can easily be shown that the projected shape on the sphere also undergoes the same rotational motion. This means that the invariants to rotation in 3D space are also invariant if the considered points are projected onto the unit sphere. The decoupled control we propose is based on this invariance property. This important property will be used to select features invariant to rotations in order to control the 3 translational DOF. In this way, the following invariant polynomials to rotations have been proposed in [31] to control the translational DOF:

$$I_1 = m_{200}m_{020} + m_{200}m_{002} + m_{020}m_{002} - m_{110}^2 - m_{101}^2 - m_{011}^2, \tag{13.12}$$

$$\begin{aligned} I_2 = &- m_{300}m_{120} - m_{300}m_{102} + m_{210}^2 - m_{210}m_{030} - m_{210}m_{012} + m_{201}^2 - m_{201}m_{021} \\ &- m_{201}m_{003} + m_{120}^2 - m_{120}m_{102} + 3m_{111}^2 + m_{102}^2 - m_{030}m_{012} + m_{021}^2 - m_{021}m_{003} \\ &+ m_{012}^2, \end{aligned} \tag{13.13}$$

$$\begin{aligned} I_3 = &\, m_{300}^2 + 3m_{300}m_{120} + 3m_{300}m_{102} + 3m_{210}m_{030} + 3m_{210}m_{012} + 3m_{201}m_{021} \\ &+ 3m_{201}m_{003} + 3m_{120}m_{102} - 3m_{111}^2 + m_{030}^2 + 3m_{030}m_{012} + 3m_{021}m_{003} + m_{003}^2. \end{aligned} \tag{13.14}$$

The invariants (13.13) and (13.14) are of higher orders than (13.12). They are thus more sensitive to noise [23, 32]. For this reason, I_1 will be used in this chapter to control the translational DOF. Therefore, the set of points has to be separated in at least three subsets to get three independent values of I_1, which allows controlling the 3 translational DOF. Furthermore, in order to decrease the variations of the interaction with respect to depth distribution, it is $s_I = \frac{1}{\sqrt{I_1}}$ that will be used instead of I_1. This will be explained in the following.

13.4.2 Variation of the Interaction Matrix with respect to the Camera Pose

As it was mentioned above, one of the goals of this work is to decrease the non-linearity by selecting adequate features. In this way, the invariance property allows us to setup some interaction matrix entries to 0. These entries will thus be constant during the servoing task. However, the other entries depend on the camera pose as it will be shown in the following. It will be also shown that the feature choice $s_I = \frac{1}{\sqrt{I_1}}$ allow obtaining interaction matrices almost constant with respect to the depth distribution.

13.4.2.1 Variation with respect to Rotational Motion

Let us consider two frames $\mathcal{F}_\mathbf{1}$ and $\mathcal{F}_\mathbf{2}$ related to the unit sphere with different orientations ($^1\mathbf{R}_2$ is the rotation matrix between the two frames) but with the same center. In this case, the value of I_t is the same for the two frames, since it is invariant to rotational motions. Let $\mathcal{X}$ and $\mathcal{X}' = {}^2\mathbf{R}_1\mathcal{X}$ be the coordinates of a projected point in the frame $\mathcal{F}_\mathbf{1}$ and $\mathcal{F}_\mathbf{2}$ respectively. Let us consider a function invariant to rotations $f(\mathcal{X}_1,\ldots,\mathcal{X}_N)$ that can be computed from the coordinates of N points onto the unit sphere (such as the invariants computed from the projection onto the unit sphere). The invariance condition between the frames $\mathcal{F}_\mathbf{1}$ and $\mathcal{F}_\mathbf{2}$ can thus be written as follow:

$$f(\mathcal{X}'_1,\ldots,\mathcal{X}'_N) = f({}^2\mathbf{R}_1\mathcal{X}_1,\ldots,{}^2\mathbf{R}_1\mathcal{X}_N) = f(\mathcal{X}_1,\ldots,\mathcal{X}_N). \tag{13.15}$$

The interaction matrix that links the variation of the function f with respect to translational velocities can be obtained as follow:

$$\mathbf{L}_{\mathbf{f_v}} = \frac{\partial f(\mathcal{X}_{s_1}+\mathbf{T},\ldots,\mathcal{X}_N+\mathbf{T})}{\partial \mathbf{T}}, \tag{13.16}$$

where $\mathbf{T}$ is a small translational motion vector. Let us now apply this formula for the camera position defined by the frame $\mathcal{F}_\mathbf{2}$:

$$\mathbf{L}'_{\mathbf{f_v}} = \frac{\partial f}{\partial \mathbf{T}} = \frac{\partial f(\mathcal{X}'_1+\mathbf{T},\ldots,\mathcal{X}'_N+\mathbf{T})}{\partial \mathbf{T}} = \frac{\partial f({}^2\mathbf{R}_1\mathcal{X}_1+\mathbf{T},\ldots,{}^2\mathbf{R}_1\mathcal{X}_N+\mathbf{T}))}{\partial \mathbf{T}}. \tag{13.17}$$

From (13.17), it can be obtained that:

$$\mathbf{L}'_{\mathbf{f_v}} = \frac{\partial f\left({}^2\mathbf{R}_1(\mathcal{X}_1+{}^1\mathbf{R}_2\mathbf{T}),\ldots,{}^2\mathbf{R}_1(\mathcal{X}_N+{}^1\mathbf{R}_2\mathbf{T})\right)}{\partial \mathbf{T}}. \tag{13.18}$$

Combining this equation with the invariance to rotations condition (13.15), we get:

$$\mathbf{L}'_{\mathbf{f_v}} = \frac{\partial f(\mathcal{X}_1 + {}^1\mathbf{R}_2\mathbf{T}, \ldots, \mathcal{X}_N + {}^1\mathbf{R}_2\mathbf{T})}{\partial \mathbf{T}}. \tag{13.19}$$

From this, we easily obtain:

$$\mathbf{L}'_{\mathbf{f_v}} = \frac{\partial f(\mathcal{X}_1 + \mathbf{T}', \ldots, \mathcal{X}_N + \mathbf{T}')}{\partial \mathbf{T}'} \frac{\partial \mathbf{T}'}{\partial \mathbf{T}}, \tag{13.20}$$

where $\mathbf{T}' = {}^1\mathbf{R}_2\mathbf{T}$. Finally, combining (13.20) with (13.16), we obtain:

$$\mathbf{L}'_{\mathbf{f_v}} = \mathbf{L}_{\mathbf{f_v}} \, {}^1\mathbf{R}_2. \tag{13.21}$$

This result was expected since applying a translational velocity $\mathbf{v_1}$ to the frame $\mathcal{F}_1$ is equivalent to applying a translational velocity to the frame $\mathcal{F}_2$ but taking into account the change of frame ($\mathbf{v_2} = {}^\mathbf{1}\mathbf{R_2}\mathbf{v_1}$). This variation is thus natural, since the translational velocities to apply to the camera frame have to depend on its orientation. Finally, this result shows that rotational motions do not change the rank of the interaction matrix of the features used to control the translational DOF. In other words, the rotational motions do not introduce singularities on the interaction matrix and any rank change of the latter depends only on the translational motions.

13.4.2.2 Variation of the Interaction Matrix with respect to Depth

Obtaining constant interaction matrix entries means that the selected features depend linearly of the corresponding DOF. In this way, in [18, 30], it was shown that for good z-axis closed-loop behavior in IBVS, one should choose image features that scale as $s \sim Z$ (Z is the object depth). This means that the variation with respect to depth is a constant (i.e. the system is linear). In the case where the object is defined by an image region, the following feature has been proposed in [30] to control the motions along the optical axis:

$$s_r = \frac{1}{\sqrt{m_{00}}},$$

where m_{00} is the bidimensional moment of order 0 (that is the object surface in the image) using the conventional perspective projection model. In the case where the object is defined by a set of discrete points, the selected optimal feature was:

$$s_d = \frac{1}{\sqrt{(\mu_{20} + \mu_{02})}}, \tag{13.22}$$

where μ_{ij} are the central moments computed from a set of discrete points (see [30] for more details). Unfortunately, s_r and s_d allows only obtaining invariance to rotations around the optical axis and not to all 3D rotations. For this reason, $s_I = \frac{1}{\sqrt{I_1}}$ will be used instead of s_r. To explain the choice of $s_I = \frac{1}{\sqrt{I_1}}$, let us first determine how the polynomial invariant I_1 behaves when Z increases by considering each term

of its formula. Let us consider the definition of the projection onto the unit sphere:

$$\begin{cases} x_s = \frac{X}{\sqrt{X^2+Y^2+Z^2}} \\ y_s = \frac{Y}{\sqrt{X^2+Y^2+Z^2}} \\ z_s = \frac{Z}{\sqrt{X^2+Y^2+Z^2}}. \end{cases} \tag{13.23}$$

From the definition of the projection onto the unit sphere, it can be obtained that if the depth Z increases (i.e. $X \ll Z$ and $Y \ll Z$), the point projection coordinates have the following behaviors with respect to depth: $x_s \sim \frac{1}{Z}$, $y_s \sim \frac{1}{Z}$ and $z_s \sim 1$. It follows that: $m_{200} = \sum_{h=1}^{N} x_{s_h}^2 \sim \frac{1}{Z^2}$, $m_{020} = \sum_{h=1}^{N} y_{s_h}^2 \sim \frac{1}{Z^2}$, $m_{110} = \sum_{h=1}^{N} x_{s_h} y_{s_h} \sim \frac{1}{Z^2}$, $m_{101} = \sum_{h=1}^{N} x_{s_h} z_{s_h} \sim \frac{1}{Z}$, $m_{011} = \sum_{h=1}^{N} y_{s_h} z_{s_h} \sim \frac{1}{Z}$ and $m_{002} = \sum_{h=1}^{N} z_{s_h}^2 \sim N$. By neglecting the term depending on $\frac{1}{Z^4}$ when the depth increases enough, the polynomial can be approximated as follow:

$$I_1 \approx N(m_{200} + m_{020}) - m_{100}^2 - m_{010}^2, \tag{13.24}$$

where N is the number of points. Therefore, it can be obtained that $I_1 \sim \frac{1}{Z^2}$ and $s_I = \frac{1}{\sqrt{I_1}} \sim Z$. Note that if the set of points is centered with respect to the optical axis (i.e. $m_{100} = m_{010} = 0$), we have:

$$I_1 \approx N(m_{200} + m_{020}). \tag{13.25}$$

In this case, note the similarity between $s_I = \frac{1}{\sqrt{I_1}}$ and the features given by (13.22). In geometrical terms, if the set of points is centered with respect to the optical axis, the projection onto unit sphere and the projection onto a classical perspective behave in the same way when the depth increases. Besides, an example of interaction matrix variations with respect to depth distributions is given in Section 13.5.1.

13.4.3 Features Selection

We could consider the center of gravity of the object's projection onto the unit sphere to control the rotational DOF:

$$\mathbf{x_{s_g}} = \left(x_{s_g}, y_{s_g}, z_{s_g} \right) = \left(\frac{m_{100}}{m_{000}}, \frac{m_{010}}{m_{000}}, \frac{m_{001}}{m_{000}} \right).$$

However, only two coordinates of $\mathbf{x_{s_g}}$ are useful for the control since the point projection belongs to the unit sphere making one coordinate dependent of the others. That is why in order to control rotation around the optical axis, the mean orientation of all segments in the image is used as a feature. Each segment is built using two different points in an image obtained by re-projection to a conventional perspective plane.

Finally, as mentioned previously, the invariants to 3D rotation $s_I = \frac{1}{\sqrt{I_1}}$ are considered to control the translation. In practice, three separate set of points such that their centers are noncollinear can be enough to control the 3 translational DOF. In order to ensure the nonsingularity of the interaction matrix, the set of points is divided in four subsets (each subset has to encompass at least 3 points). This allows us to obtain four different features to control the 3 translational DOF.

13.5 Results

In this section, an example of interaction matrix variations with respect to depth distribution is given. Thereby, several results of pose estimation and visual servoing are presented.

13.5.1 Variation of the Interaction Matrix with respect to Depth Distribution

Fig. 13.3 gives the variations of the interaction matrix entries of I_1 and $s_I = \frac{1}{\sqrt{I_1}}$ with respect to translational motion applied along the optical axis to the four random coplanar points defined in the camera frame as follow:

$$\mathbf{X_o} = \begin{pmatrix} -0.3258 & -0.0811 & 0.1487 & 0.2583 \\ -0.0458 & 0.1470 & -0.1052 & 0.0039 \\ 1.0000 & 1.0000 & 1.0000 & 1.0000 \end{pmatrix}. \tag{13.26}$$

The set of points has been chosen to be approximatively centered with respect to the $z-axis$ ($m_{100} \approx 0\ m_{010} \approx 0$). For this reason, it can be seen that $L_x \approx L_{x_1} \approx L_y \approx L_{y_1} \approx 0$ ($\mathbf{L}_{I_1} = \left[L_x, L_y, L_z, 0, 0, 0\right]$ and $\mathbf{L}_{s_I} = \left[L_{x_1}, L_{y_1}, L_{z_1}, 0, 0, 0\right]$). In practice, the features I_1 and s_I also depend mainly on the translational motion with respect to the object axis of view. From Fig. 13.3(a) and Fig. 13.3(b), it can be seen that L_{z_1} is almost constant and largely invariant to the object depth. On the other hand L_z decreases to 0 when the object depth increases.

13.5.2 Visual Servoing Results

In these simulations, the set of points is composed of 4 noncoplanar points. For all the following simulations, the desired position corresponds to the 3D points coordinates defined in the camera frame as follow:

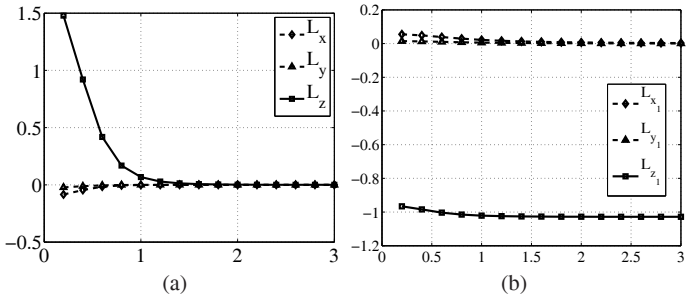

Fig. 13.3 Variations of the interaction matrix with respect to depth distribution (in meter): (a) results for I_1; and (b) results for $s_I = \frac{1}{\sqrt{I_1}}$.

$$\mathbf{X_d} = \begin{pmatrix} 0 & -0.2 & 0 & 0.2 \\ 0.2 & 0 & -0.2 & 0 \\ 0.9 & 1. & 1 & 1.2 \end{pmatrix}. \tag{13.27}$$

From the four set of points 4 different triangles can be obtained. For each triangle, the invariant $s_I = \frac{1}{\sqrt{I_1}}$ is computed to control the translational motion.

In a first simulation, we show the advantage of using $s_I = \frac{1}{\sqrt{I_1}}$ instead of using directly I_1. For this purpose, the translational motion given by (13.28) has been considered between the desired and the initial camera poses. Further, in the control law (13.4), the scalar λ that tunes the velocity has been set to 1 and the interaction matrix computed at each iteration is used (i.e. $\widehat{\mathbf{L_s}} = \mathbf{L_s}$). If the system were completely linear, the convergence would be obtained in only one iteration. The nonlinearity of the system has as effect to damp or to magnify the camera velocities. In our case (i.e $\lambda = 1$), the nonlinearity can slow the convergence (damping the velocity) or it can produce oscillations (magnifying the velocity). The results obtained using $I_t = \frac{1}{\sqrt{I_1}}$ and using I_1 are given on Fig. 13.4. From Fig. 13.4(a) and Fig. 13.4(b), oscillations can be observed for the features errors as well as for the velocities obtained using I_1 before converging (after 9 iterations). On the other hand, a fast convergence is obtained using $I_t = \frac{1}{\sqrt{I_1}}$ without oscillations (after only two iterations the system has almost converged). This shows that using $I_t = \frac{1}{\sqrt{I_1}}$, the system behaved almost as a linear system.

$$\mathbf{t_0} = \begin{pmatrix} 0.2, & 0.3, & 0.6 \end{pmatrix} \text{ m}. \tag{13.28}$$

In a second simulation, the rotational motion defined by the rotation vector (13.29) has been considered. The rotation matrix is obtained from the rotation vector $\theta\mathbf{u}$ using the well known Rodrigues formula. We compare the system behavior using our features and using the point Cartesian coordinates (a conventional perspective projection has been considered). The obtained results are given on Fig. 13.5. From Fig. 13.5(a), it can be seen that a nice decrease of the features errors is obtained using our features. Furthermore, from Fig. 13.5(b), since the considered translational

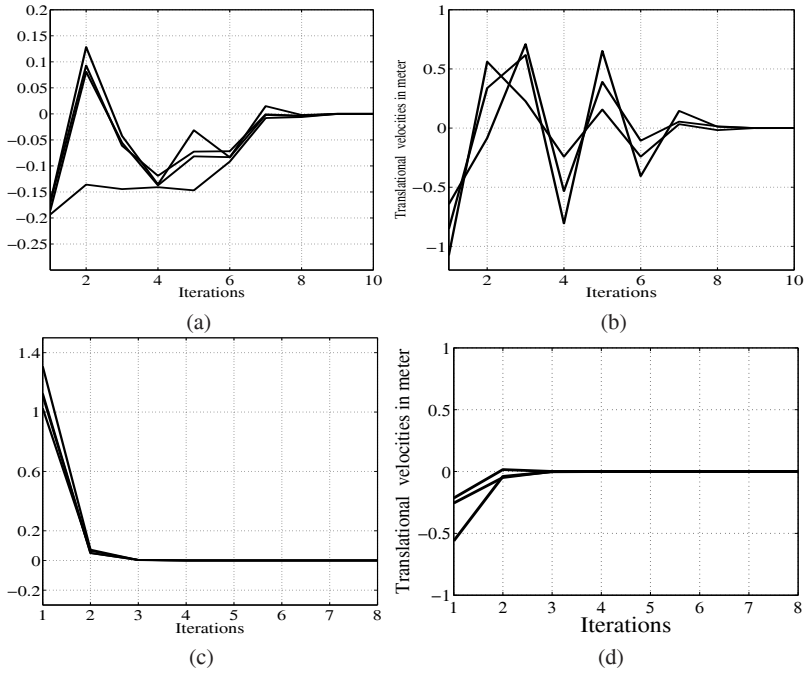

Fig. 13.4 Results using I_1: (a) features errors; (b) velocities (in m/s). Using $s_I = \frac{1}{\sqrt{I_1}}$: (c) features errors; and (d) velocities (in m/s).

motion is null, the translational velocity computed using the invariants to rotations are null (thanks to the invariance to rotations). Further, as for feature errors, Fig. 13.5(c) shows a nice decrease of the rotational velocities. The results obtained using the point Cartesian coordinates to control the camera position are given on Fig. 13.5(d), Fig. 13.5(e) and Fig. 13.5(f). Fig. 13.5(d) shows a nice decrease of the feature errors. On the other hand, the behavior of the velocities is far from satisfactory. Indeed, a strong translational motion is observed (see Fig. 13.5(e)) and since the rotational DOF are coupled with the translational one, this introduced also a strong oscillations of the rotational velocities (see Fig. 13.5(f)).

$$\boldsymbol{\theta}\mathbf{u} = \left(-6.42,\ 19.26,\ 128.40\right) \text{ deg.} \tag{13.29}$$

For the third simulation, the motion between the initial and desired camera poses defined by (13.30) and (13.30) has been considered. The same desired camera pose as for the first experiment was used. From Fig. 13.6(a), it can be noticed that the feature errors behavior is very satisfactory. The same satisfactory behavior is obtained for translational and rotational velocities (see Fig. 13.6(b) and Fig. 13.6(c)). Indeed, nice decreases of the feature errors as well as for the velocities are obtained. On the other hand the results obtained using the point Cartesian coordinates show a strong translational motion generated by the wide rotation and also oscillations of

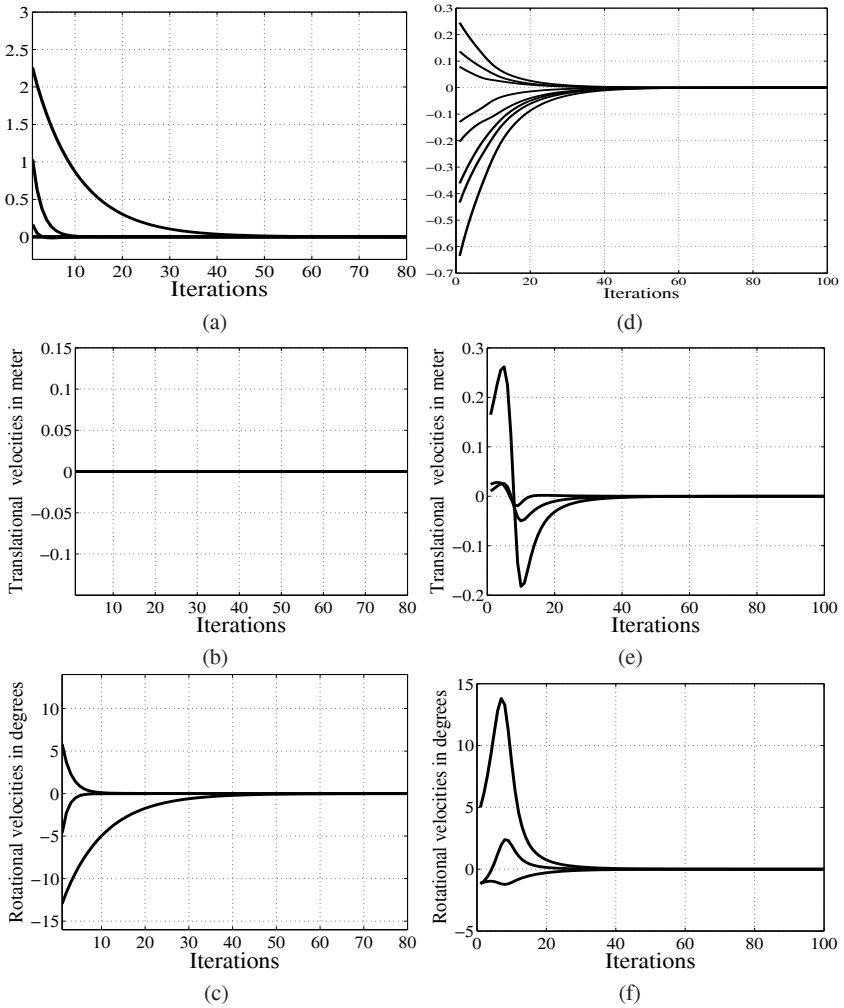

Fig. 13.5 Results for pure rotational motion. Using $S_I = \frac{1}{\sqrt{I_1}}$: (a) features errors; (b) translational velocities (in m/s); (c) rotational velocities (in deg/s). Using point coordinates: (d) features errors; (e) Translational velocities (in m/s); and (f) rotational velocities (in deg/s).

the whole velocities (see Fig. 13.6(e) and Fig. 13.6(f))

$$\boldsymbol{\theta u} = \left(-6.42,\ 19.26,\ 128.40\right) \text{deg} \mathbf{t_1} = \left(-0.,\ -0.3,\ 1\right) \text{m}. \tag{13.30}$$

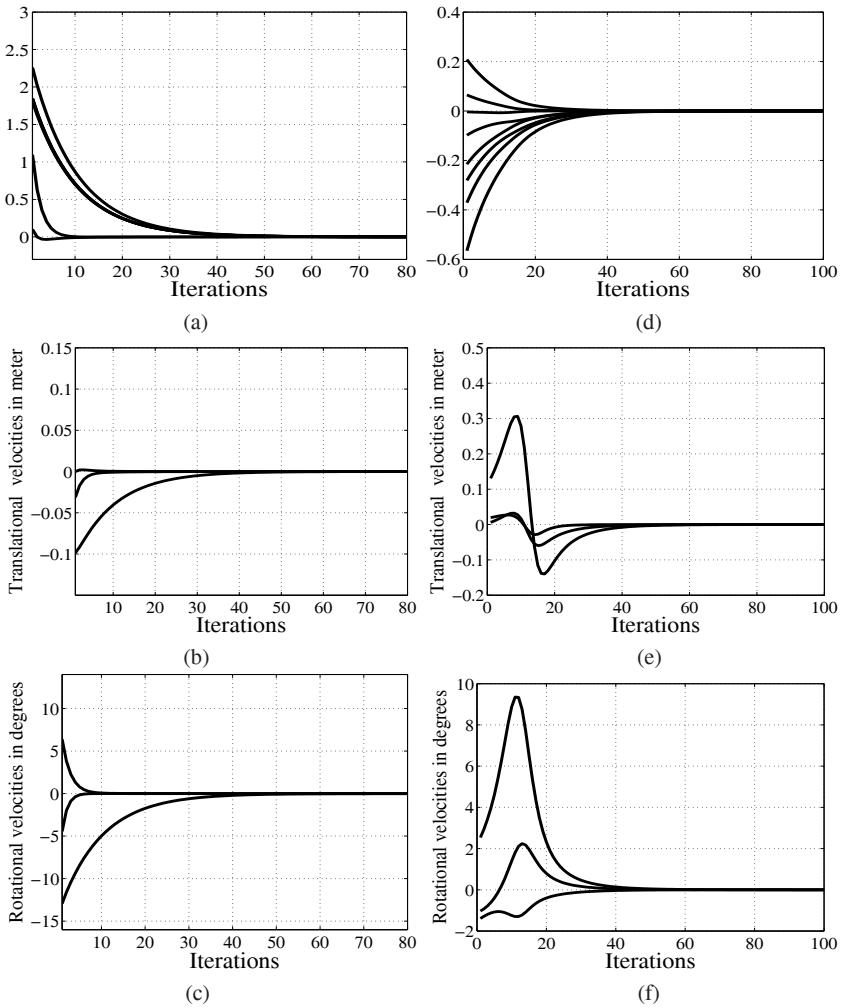

Fig. 13.6 Results for general motion. Using $I_t = \frac{1}{\sqrt{I_t}}$: (a) features errors; (b) translational velocities (in m/s); (c) rotational velocities (in deg/s). Using point coordinates: (d) features errors; (e) translational velocities (in m/s); and (f) rotational velocities (in deg/s).

13.5.3 Pose Estimation Results

In this part, our pose estimation method is compared with the linear method proposed by Ansar in [1] and the iterative method proposed by Araujo [2]. The identity matrix has been used to initialize ${}^{i}\mathbf{M}_o$ for our method and for the iterative method proposed in [2]. The combination of the linear method and iterative method proposed by Araujo is also tested. In other words, the results obtained by the linear

method will be used as initialization to the iterative method. The following setup has been used:

- an object composed of four points forming a square defined as follows has been considered:
$$X_o = \begin{bmatrix} -0.2 & 0.2 & -0.2 & 0.2 \\ -0.2 & -0.2 & 0.2 & 0.2 \\ 1. & 1. & 1. & 1 \end{bmatrix};$$
- a camera model with focal $F = 800$ and principal point coordinates $u = v = 400$ pixels has been used to compute the points coordinates in image;
- the interaction matrix corresponding to the current position is used in the control law (13.4) to compute the camera displacement (i.e $\widehat{\mathbf{L}_\mathbf{s}} = \mathbf{L}_\mathbf{s}$);
- random poses have been generated as follow:
  - 1000 random translational motions $\mathbf{t} = (1\sigma_1 \;\; 1\sigma_2 \;\; 1.3\sigma_3)$ are firstly applied to the point coordinates defined in the square frame, where σ_1 and σ_2 are random numbers chosen from a normal distribution with mean zero, variance one and standard deviation one, σ_3 is a random number chosen from a uniform distribution on the interval [0.0 1.0];
  - the rotational motion is chosen such that the points coordinates belongs to the image limits [1 800; 1 800]. Further, the rotational motion with respect to the optical axis can range randomly between $[0 \;\; 2\pi]$.

Let us consider the pose error defined by:

$$\mathbf{T_e} = \begin{pmatrix} \mathbf{R_e} & \mathbf{t_e} \\ \mathbf{0_{1\times 3}} & 1 \end{pmatrix} = \mathbf{T_r^{-1}T_c},$$

where $\mathbf{T_r}$ and $\mathbf{T_c}$ are respectively the real and the estimated pose. If the correct pose is obtained, $\mathbf{T_e}$ is equal to the identity matrix ($\| \mathbf{t_e} \| = 0$ and $\mathbf{R_e} = \mathbf{I_3}$). Let θ_e be the rotation error corresponding to the rotation matrix $\mathbf{R_e}$. Fig. 13.7 and 13.8 give the distribution of θ_e and $\| \mathbf{t_e} \|$ using the four different methods and for three different levels of noise. In other words, for each values of θ_e and $\| \mathbf{t_e} \|$, the plot gives the percentage of the errors smaller or equal to these values.

Fig. 13.7(a) and Fig. 13.8(a) give the distributions of θ_e and $\| \mathbf{t_e} \|$ when perfect data is considered (without noise and using the exact camera parameters). From these figures, it can be seen that the linear method, our method and **AA** method have always estimated the exact pose. On the other hand, Araujo's method initialized by the identity matrix only converges for nearly 40% cases. Fig. 13.7(b) and Fig. 13.8(b) give the obtained results with 0.3 pixels gaussian noises, principal points coordinates [375 375] and focal $F = 760$ (recall that the real values are [400 400] for the principal point and $F = 800$ for the focal). From these figures, it can be noticed that the accuracy of the estimation using the linear method decreases when the data noise increases. The results obtained using the linear method are improved using Araujo's method. On the other hand, the accuracy of the Araujo iterative method initialized by the identity matrix also decreases, but the convergence percentage is

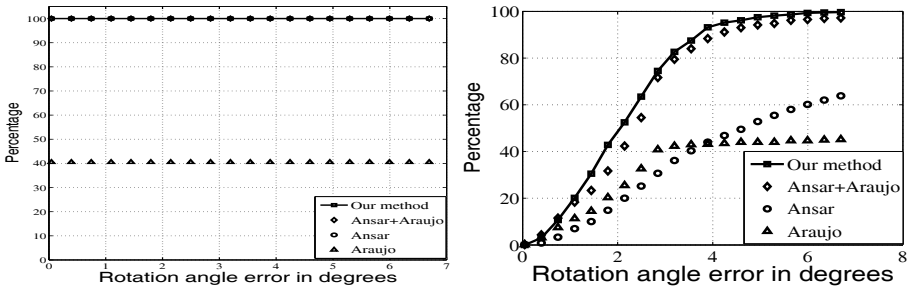

Fig. 13.7 Distribution of θ_e with respect to noise level and errors on camera parameters: (a) results with zero noise and exact camera; and (b) results for 0.3 pixels gaussian noise, principal points coordinates [375 375] and focal $F = 760$.

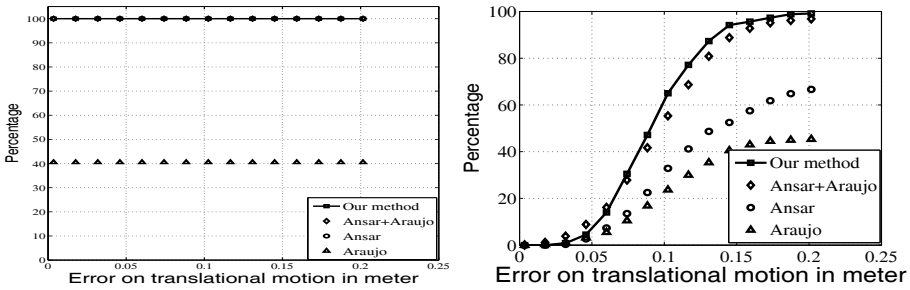

Fig. 13.8 Distribution of $\| \mathbf{t_e} \|$ with respect to noise level and errors on camera parameters: (a) results with zero noise; and (b) results for 0.3 pixels gaussian noise, principal points coordinates [375 375] and focal $F = 760$.

still around 40%. Finally for the same experiment, our iterative method converges for all cases and gives more accurate estimation of the poses.

13.6 Conclusion

In this chapter, a unique and efficient decoupled scheme for visual servoing and pose estimation has been proposed. The proposed scheme is valid for cameras obeying the unified model. More precisely, the invariants to rotational motions computed from the projection onto the unit sphere are used to control the translational DOF. Adequate forms of invariants have been proposed to decrease the interaction matrix variations with respect to the depth distributions.The validations results have shown the efficiency of the proposed scheme. Future works will be devoted to extend these results to model-free pose estimation problem and to region-based visual servoing.

References

[1] Ansar A, Daniilidis K (2003) Linear pose estimation from points or lines. IEEE Trans on Pattern Analysis and Machine Intelligence 25:282–296
[2] Araujo H, Carceroni RL, Brown CM (1998) A fully projective formulation to improve the accuracy of lowe's pose-estimation algorithm. Computer Vision and Image Understanding 70:227–238
[3] Baker S, Nayar S (1999) A theory of catadioptric image formation. Int Journal of Computer Vision 35(2):175–196
[4] Barreto J, Araujo H (2001) Issues on the geometry of central catadioptric image formation. In: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol 2, pp II–422–II–427 vol.2
[5] Chaumette F (1998) Potential problems of stability and convergence in image-based and position-based visual servoing. In: Kriegman D, Hager G, Morse A (eds) The Confluence of Vision and Control, LNCIS Series, No 237, Springer-Verlag, pp 66–78
[6] Corke PI, Hutchinson SA (2001) A new partitioned approach to image-based visual servo control. IEEE Trans on Robotics and Automation 17(4):507–515
[7] Dementhon D, Davis L (1995) Model-based object pose in 25 lines of code. Int Journal of Computer Vision 15(1-2):123–141
[8] Dhome M, Richetin M, Lapreste JT, Rives G (1989) Determination of the attitude of 3d objects from a single perspective view. IEEE Trans on Pattern Analysis and Machine Intelligence 11(12):1265–1278
[9] Espiau B, Chaumette F, Rives P (1992) A new approach to visual servoing in robotics. IEEE Trans on Robotics and Automation 8:313–326
[10] Fiore P (2001) Efficient linear solution of exterior orientation. IEEE Trans on Pattern Analysis and Machine Intelligence 23(2):140–148
[11] Fomena R, Chaumette F (2007) Visual servoing from spheres using a spherical projection model. In: Robotics and Automation, 2007 IEEE International Conference on, pp 2080–2085
[12] Geyer C, Daniilidis K (2003) Mirrors in motion: Epipolar geometry and motion estimation. Int Journal on Computer Vision 45(3):766–773
[13] Hamel T, Mahony R (2002) Visual servoing of an under-actuated dynamic rigid body system: an image-based approach. IEEE Trans on Robotics and Automation 18(2):187–198
[14] Iwatsuki M, Okiyama N (2002) A new formulation of visual servoing based on cylindrical coordinates system with shiftable origin. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Lausanne, Switzerland, pp 354–359
[15] Lee JS, Suh I, You BJ, Oh SR (1999) A novel visual servoing approach involving disturbance observer. In: IEEE Int. Conf. on Robotics and Automation, ICRA'99, Detroit, Michigan, pp 269–274
[16] Lowe DG (1991) Fitting parameterized three-dimensional models to images. IEEE Trans on Pattern Analysis and Machine Intelligence 13:441–450

[17] Lu CP, Hager G, Mjolsness E (2000) Fast and globally convergent pose estimation from video images. IEEE Trans on Pattern Analysis and Machine Intelligence 22(6):610–622

[18] Mahony R, Corke P, Chaumette F (2002) Choice of image features for depth-axis control in image-based visual servo control. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'02, Lausanne, Switzerland, vol 1, pp 390–395

[19] Malis E, Chaumette F, Boudet S (1999) 2 1/2 d visual servoing. IEEE Trans on Robotics and Automation 15(2):238–250

[20] Marchand E, Chaumette F (2002) Virtual visual servoing: A framework for real-time augmented reality. In: Drettakis G, Seidel HP (eds) EUROGRAPHICS 2002 Conference Proceeding, Saarebrücken, Germany, Computer Graphics Forum, vol 21(3), pp 289–298

[21] Martinet P, Gallice J (1999) Position based visual servoing using a nonlinear approach. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'99, Kyongju, Korea, vol 1, pp 531–536

[22] Mei C, Rives P (2007) Single view point omnidirectional camera calibration from planar grids. In: Robotics and Automation, 2007 IEEE International Conference on, pp 3945–3950

[23] Mukundan R, Ramakrishnan KR (1998) Moment Functions in Image Analysis Theory and Application. M. World Scientific Publishing, Co.Pte.Ltd

[24] Rives P, Azinheira J (2004) Linear structures following by an airship using vanishing points and horizon line in a visual servoing scheme. In: IEEE Int. Conf. on Robotics and Automation, ICRA'04, New Orleans, Louisiana, pp 255–260

[25] Safaee-Rad R, Tchoukanov I, Smith K, Benhabib B (1992) Three-dimensional location estimation of circular features for machine vision. IEEE Trans on-Robotics and Automation 8(5):624–640

[26] Sundareswaran V, Behringer R (1999) Visual servoing-based augmented reality. In: IWAR '98: Proceedings of the international workshop on Augmented reality : placing artificial objects in real scenes, A. K. Peters, Ltd., Natick, MA, USA, pp 193–200

[27] Svoboda T, Pajdla T (2002) Epipolar geometry for central catadioptric cameras. Int Journal on Computer Vision 49(1):23–37

[28] Tahri O (2004) Utilisation des moments en asservissement visuel et en calcul de pose. PhD thesis, University of Rennes

[29] Tahri O, Chaumette F (2005) Complex objects pose estimation based on image moment invariants. In: IEEE Int. Conf. on Robotics and Automation, ICRA'05, Barcelona, Spain, pp 438–443

[30] Tahri O, Chaumette F (2005) Point-based and region-based image moments for visual servoing of planar objects. IEEE Transactions on Robotics 21(6):1116–1127

[31] Tahri O, Chaumette F, Mezouar Y (2008) New decoupled visual servoing scheme based on invariants from projection onto a sphere. In: IEEE Int. Conf. on Robotics and Automation, ICRA'08, pp 3238–3243

[32] Teh C, Chin RT (1988) On image analysis by the method of moments. IEEE Trans on Pattern Analysis and Machine Intelligence 10(4):496–513
[33] Wilson W, Hulls C, Bell G (1996) Relative end-effector control using cartesian position-based visual servoing. IEEE Trans on Robotics and Automation 12(5):684–696